

## ImageJ-suodin käyttäen FFT:tä

Suodatusta voidaan tehdä Fourier-tasossa. Olkoon  $H$  haluttu suodattimen taajuusvaste,  $F$  kuvan Fourier-muunnos. Tällöin suodatettu kuva  $G = F H$ . Taajuustason kertolasku vastaa aikatason konvoluutiota. Ideaalisen, jyrkkäreunaisen taajuusvasteen käyttämiseen liittyy ongelmia, mutta yksinkertaisuuden vuoksi nyt tehdään sellainen toteutus ImageJ:llä. Tarkempaa kuvausta Fourier-tason suodatuksesta löytyy esimerkiksi osoitteesta:

<http://homepages.inf.ed.ac.uk/rbf/HIPR2/freqfilt.htm>

Tehdään ImageJ-plugin käyttäen valmiita FFT- ja InverseFFT-komentoja. ImageJ-pluginin lähdekoodi:

```
import ij.*;
import ij.process.*;
import ij.gui.*;
import java.awt.*;
import ij.plugin.filter.*;

// Tekee kuvalle DFT:n, muokkaa Fourier-muunnosta halutulla taajuusvasteella ja tekee
// käänteismuunnoksen
// Anna syötteenä neliönmuotoinen kuva, jonka sivun pituus on 2^N
public class DFT_filter_ implements PlugInFilter {
    ImagePlus imp;

    public int setup(String arg, ImagePlus imp) {
        this.imp = imp;
        return DOES_8G;
    }

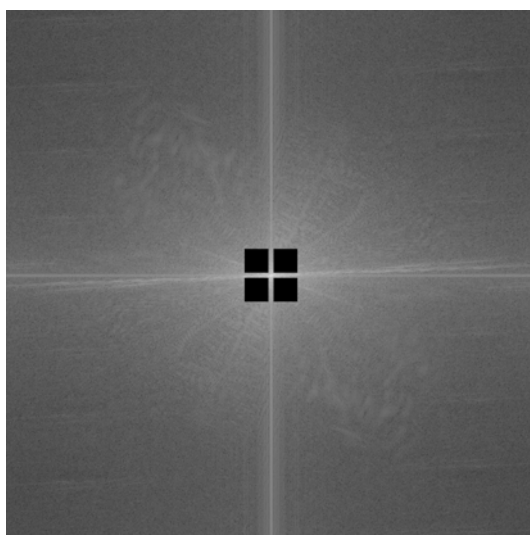
    public void run(ImageProcessor ip)
    {
        int w = ip.getWidth();

        // Suodatuksen taajuusrajat (100 % näytteenottotaajuudesta)
        int alaraja = 1;
        int ylaraja = 10;
        // Taajuusrajat pikseleinä
        int lb = (alaraja*w/2)/100;
        int ub = (ylaraja*w/2)/100;
        int L = ub-lb;

        IJ.run("FFT");
        ImagePlus imp2 = IJ.getImage();
        imp2.show();
        ImageProcessor ip2 = imp2.getProcessor();
        // Nollataan halutut taajuudet symmetrisesti jokaisesta neljänneksestä
        ip2.setColor(0);
        // Vasen yläneljännes
        IJ.makeRectangle(w/2-lb-L, w/2-lb-L, L, L);
        ip2.fill();
        // Oikea yläneljännes
        IJ.makeRectangle(w/2+lb, w/2-lb-L, L, L);
        ip2.fill();
        // Vasen alaneljännes
        IJ.makeRectangle(w/2-lb-L, w/2+lb, L, L);
        ip2.fill();
        // Oikea alaneljännes
        IJ.makeRectangle(w/2+lb, w/2+lb, L, L);
        ip2.fill();
        IJ.makeRectangle(0, 0, 0, 0);
        IJ.run("Inverse FFT");
    }
}
```



Kuva 1. Alkuperäinen.



Kuva 2. Fourier-muunnos, kun taajuudet  $[1, 10]$  % näytteenottotaajuudesta on nollattu.



Kuva 3. Suodatettu kuva.